

Insurance Cost Estimator

A machine learning web application for predicting insurance charges

By: Delilah Slabaugh

What the App Does

Takes user input for age, sex, BMI, children, smoker status, and region

Uses a trained Ridge Regression model to predict insurance charges Displays the estimated cost based on normalized features

Model Choice



RIDGE REGRESSION SELECTED FOR ITS BALANCE BETWEEN BIAS AND VARIANCE



PERFORMED WELL ON SYNTHETIC DATA WITH MINIMAL OVERFITTING



REGULARIZATION HELPED MAINTAIN GENERALIZATION WITHOUT COMPLEX TUNING

Project Methodology

Used Kaggle's synthetic insurance.csv dataset

Preprocessing steps:

- Encoded categorical variables (e.g., smoker, sex, region)
- Normalized and cleaned numeric data

Trained a ridge regression model using scikit-learn

Evaluated using metrics like RMSE and R2

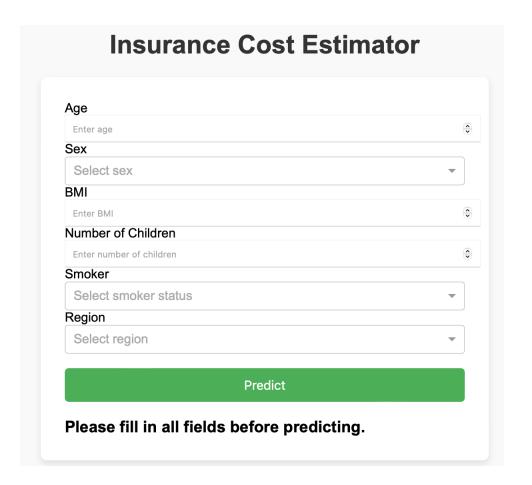
Built the web app with Dash

Initial User Interface

The user is presented with a simple UI containing fields for:

- Age
- Sex
- BMI
- Number of Children
- Smoker status
- Region

Once these fields are filled, the user should click "Predict" to generate an estimated cost.



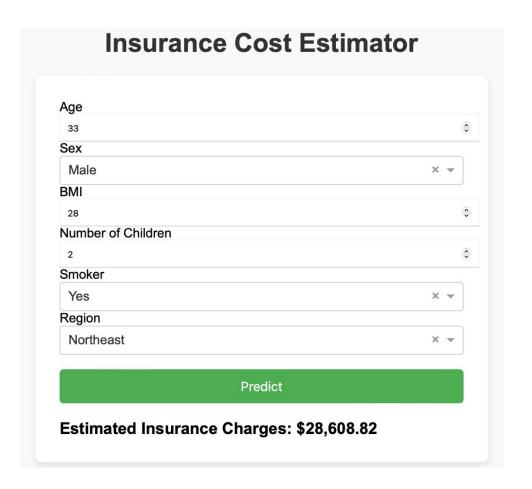
Young Male, Smoker, BMI 28, 2 Children, Northeast

Prediction:

Estimated Insurance Charges: \$28,608.82

• Interpretation:

Smoking drastically in creases the cost. Even without obesity or age-related risks, smoking is the dominant factor here.



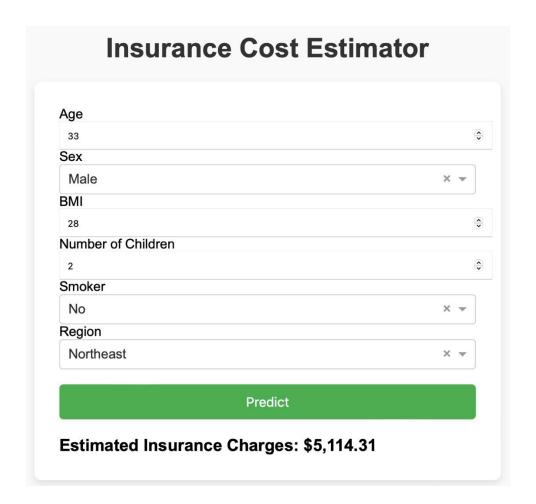
Young Male, BMI 32, 1 Child, Non-Smoker, Northwest

• Prediction:

Estimated Insurance Charges: \$5,114.31

• Interpretation:

Just by switching from smoker to non-smoker, and decreasing by 1 child, the cost drops by over \$23,000. This shows the model heavily weighs smoking status in risk scoring.



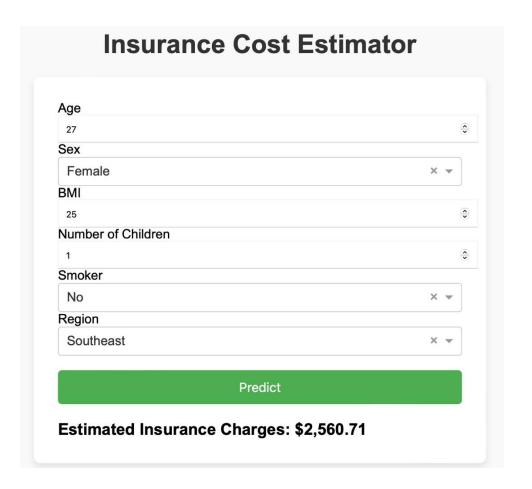
Young Female, BMI 25, 1 Child, Non-Smoker, Southeast

Prediction:

Estimated Insurance Charges: \$2,560.71

• Interpretation:

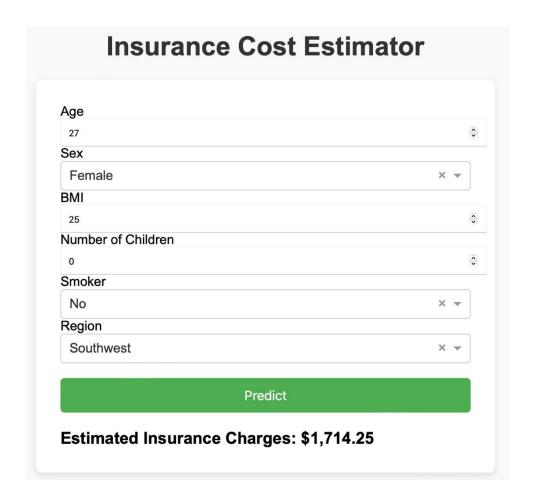
Costs are significantly lower for a healthy, non-smoking female. Region also plays a role in reducing costs.



Young Female, BMI 25, 0 Children, Non-Smoker, Southwest

- Prediction:

 Estimated Insurance
 Charges: \$1,714.25
- Interpretation:
 Removing
 dependents and changing the region slightly lowers the premium further.



What I Learned & Next Steps

Gained hands-on experience building a full ML pipeline

Learned how to use Dash to create interactive apps

Next steps:

Deploy app on Render or GitHub Pages with Dashto-HTML export

Add more robust error handling and logging Retrain model using real-world or expanded datasets

Make the UI mobile responsive

Add data visualizations for further insights

Insurance Cost Estimator Project Write-Up

This application is a simple insurance cost estimator built using Dash. Users can enter basic info like age, sex, BMI, number of children, smoking status, and region, and the app returns an estimated insurance charge based on those inputs. The goal was to create something that's easy to use and gives a quick result, even for people without a data background.

I chose Ridge Regression as the model because it handles overlapping features better than basic linear regression and keeps things interpretable. I tested a few other models, but Ridge gave consistent results without overfitting. It also runs fast and works well with scaled data, which made it a good choice for a lightweight application like this.

The project started with cleaning and exploring the insurance.csv dataset. I scaled the continuous variables, one-hot encoded the categorical ones, and created new groupings for age and BMI. After evaluating models using cross-validation, I saved the best Ridge model with joblib and connected it to the app using a Dash frontend.

Next steps would include deploying the app to the web using services like Streamlit Community Cloud, PythonAnywhere, or even Docker on AWS for more control. I want to make the app more user-friendly by giving better guidance when inputs are missing or incorrect. Another priority is retraining the model using real-world data instead of synthetic data to boost accuracy and make the predictions more useful. On the backend, I'd like to add logging to help track how the app is performing and include input validation to prevent unexpected behavior and improve reliability overall.

This project helped me pull everything together, from cleaning the data and training the model to building the user interface and getting the app running. It gave me real experience turning a machine learning model into something that regular people can use, not just something that runs in a notebook. I also saw the importance of making apps that are easy to understand and interact with. I still want to get better at scaling apps for more users, handling real-time inputs, and making sure dashboards work across different devices. Those are areas I'll keep working on as I continue growing my portfolio and preparing for real-world use cases.